

AD-A160 977

A FAST ALGORITHM FOR NON-NEWTONIAN FLOW(U) WISCONSIN
UNIV-MADISON MATHEMATICS RESEARCH CENTER D S MALKUS
SEP 85 MRC-TSR-2860 DAAG29-80-C-0041

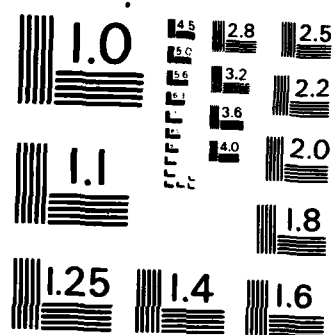
1/1

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

AD-A160 977

MRC Technical Summary Report #2860

A FAST ALGORITHM FOR NON-NEWTONIAN FLOW

David S. Malkus

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

DTIC FILE COPY

September 1985

(Received August 14, 1985)

SEP 1985
B

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

Air Force Office of
Scientific Research
Washington, DC 20332

National Science Foundation
Washington, DC 20550

85 11 06 064

UNIVERSITY OF WISCONSIN
MATHEMATICS RESEARCH CENTER
A FAST ALGORITHM FOR NON-NEWTONIAN FLOW

David S. Malkus

Technical Summary Report #2860
September 1985

ABSTRACT

The goals of the project described here are twofold: First, to turn an existing pilot algorithm for the steady flow of non-Newtonian memory fluids into a robust and efficient algorithm. Second, render enhancements of the method's current capabilities computationally feasible. Such enhancements include fully coupled thermal dependence, material compressibility, and free surface flows. The pilot algorithm is a finite element method whose novelty lies in its computation of the stress field in a nonlinear iteration scheme. The stress at a point is a non-local functional of the current velocity iterate, and the pilot method has demonstrated the feasibility of reliable computation with such constitutive equations. Before the method can take its place as a reliable scientific and engineering tool, intensive effort must be made to reduce the computational cost in the manner described here.

Application of the method to viscoelastic fluids in free flow

AMS (MOS) Subject Classifications: 65N30, 76A05, 76A10

Key Words: memory fluid, non-Newtonian fluid, finite element, vectorization,
adaptive memory quadrature, Jacobian, pseudo-dynamic relaxation

Work Unit Number 2 (Physical Mathematics) and 3 (Numerical Analysis
and Scientific Computing)

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041, the United States Air Force under Grant No. AFOSR-85-0141 and the National Science Foundation under Grant Nos. MCS 79-03542, 81-02089 and 83-01433.

A FAST ALGORITHM FOR NON-NEWTONIAN FLOW

David S. Malkus

I. VISCOELASTIC FLUIDS. The following equations are solved numerically, using the finite element method [1 - 4]: The equations of steady motion,

$$\nabla \cdot \sigma + f = \rho(u \cdot \nabla)u \quad (1)$$

where u is the velocity field, σ the stress tensor, f a body force, and ρ the density. The equation of continuity for an incompressible fluid is

$$\nabla \cdot u = 0 \quad (2)$$

For non-Newtonian fluids, the crucial equation is the constitutive equation,

$$\sigma = -pI + 2\mu(0)R\dot{e} + (1 - R)\sigma' \quad (3)$$

where p is an isotropic contribution to the stress, $\mu(0)$ is a zero-shear viscosity, R is a ratio of a retardation time, Λ , to a retardation time, T , and σ' is an extra stress tensor. The ratio, R , and its complement determine the proportion of the stress which is Newtonian — and usually is the result of a Newtonian solvent — and the complementary proportion from the extra stress — usually due to long-chain molecules (such as polymers) dissolved in the Newtonian solvent.

There are many proposed forms for the extra-stress tensor; there are two basic categories: the differential and integral models [1]. Here we shall only be concerned with the integral form.

$$\sigma' = \frac{\mu_0}{T} \sum_{l=1}^M \int_{-\infty}^0 S_0^{(l)}(\tau) m_l(\tau) d\tau \quad (4)$$

$$m_l(\tau) = T^{-1} \sum_{k=1}^N G_k^{(l)} p_k^{(l)} \exp\left(-\frac{\tau p_k^{(l)}}{T}\right)$$

where μ_0 is a constant determining $\mu(0)$, and $S_0^{(l)}$ is a strain measure, measuring the deformation which carried the particle from its position at time τ in the past to the stress evaluation point at the present time, 0. The strain measures are the same kind employed in finite elasticity. The memory functions, m_l , are usually sums of exponentials, each with amplitude determined by the modulus, $G_k^{(l)}$ and decay constant, p_k , which determines the fraction of the basic decay rate, T .

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041, the United States Air Force under Grant No. AFOSR-85-0141 and the National Science Foundation under Grant Nos. MCS79-03542, 81-02089 and 83-01433.



A-1

Thus a computational method must determine the deformation history of every stress evaluation point required to solve the equations of motion in some approximate way, compute the required strain measure — which is almost always highly nonlinear in its dependence on the velocity field — and then approximate the history integral over an infinite interval. This just computes the stress, and then the stress computation must be imbedded in some iterative scheme to produce an approximate solution to the highly nonlinear equations of motion.

II. SOME FLOWS. In this section we give a brief description of some of the flows to which the current method is being applied. The geometry of these flows is quite simple and the results obtained do not illustrate the real power of the finite element method. It is hoped that the reader will appreciate that the method described here is still very much in the development stage, and that the problems so far investigated by the author and other researchers are intended to isolate the complexity inherent in the non-Newtonian nature of the flow from other possible complications. Nevertheless, there seems to be a good deal of physical interest in the problems pictured here, in spite of their geometric simplicity.

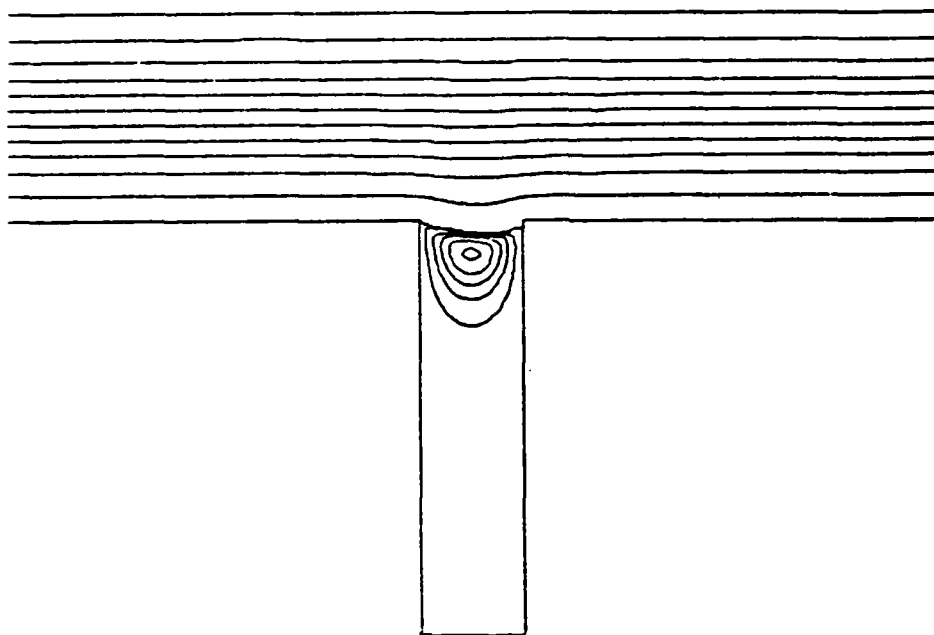


FIGURE 1

Flow over a transverse slot computed by the pilot method on a mesh of 1008 elements.

The first flow is a plane flow over a transverse slot. The streamlines plotted in Figure 1 are taken from a solution computed by the author, using a constitutive equation of his own devising [1] and the mesh of 1008 crossed-triangle macroelements illustrated in ref. 1. The flow is at a "Deborah number" of 4.7 (this can be thought of as a dimensionless shear rate). Flow is from right to left, and undisturbed flow profiles have been imposed at the

inflow and outflow. Actually, since there is fluid memory, the inlet condition is that the flow continues forever upstream as undisturbed plane Poiseuille flow. Figure 1 illustrates a characteristic tilt to the vortex in the slot, which is opposite in direction to the tilt observed in Newtonian flows with non-zero Reynolds number [3].

The interest in flows over transverse slots arises from the fact that there seems to be an important relation between the difference between the pressures at top and bottom of the slot and the first normal-stress difference of the fluid in the undisturbed flow [1 - 3]. There seems to be a discrepancy between what the numerical models predict and laboratory experiments measure in such flows, and it is one of the author's highest priorities to resolve this. The results could have important ramifications for devices designed to measure the first normal-stress difference using "hole-pressure" measurements.

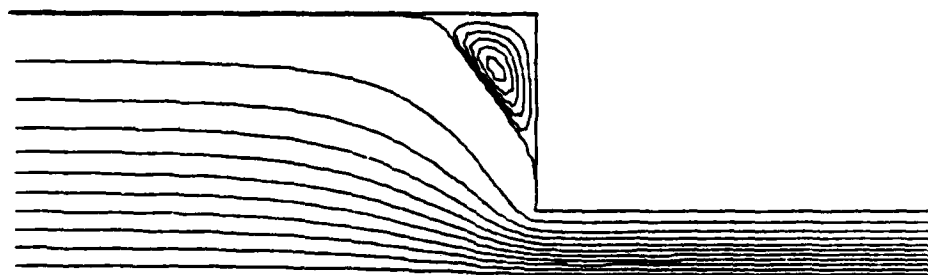


FIGURE 2

Abrupt contraction flow computed by the pilot method with 940 elements.

The next flow is that of flow through an abrupt, planar contraction. Figure 2 pictures such a flow: flow is from left to right so that the fluid is being forced from the larger to the smaller channel. Because symmetry is assumed, the computational domain is only the top half of a channel cross-section. The flow pictured here uses the same fluid model as that of Figure 1, at a slightly lower Deborah number, 3.7. Inflow and outflow conditions are imposed as before; extreme care is taken to match the flux at inflow and outflow boundaries.

The interest in this flow stems from the fact that some fluids seem to behave quite differently than others in contraction flow. Some fluids, such as polystyrene or high-density

polyethylene melts, seem to have relatively smaller "dead-spaces" or recirculation regions at high Deborah number than at low Deborah number, while some branched polymers, such as low-density polyethylene, seem to do quite the opposite, developing recirculation regions emanating from entry which dominate the whole flow-field. The flow pictured in Figure 2 has about the same size recirculation region as a flow of the same fluid at low shear rate. The author is interested in further study of this flow in order to find out what property of the constitutive model is associated with entry vortex behavior. The ability to predict recirculation size is of practical import because fluid trapped inside dead-spaces tends to degrade. It would be useful to be able to determine how much polymer degradation could be expected in a given die as a function of measurable material properties.

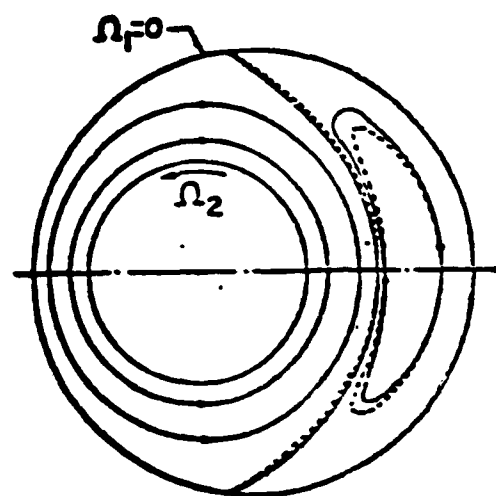


FIGURE 3

Newtonian flow in a plane cross-section of a journal-bearing.

The final flow is one for which the author has as yet no results: that is the flow in a journal-bearing. Figure 3 pictures the cross-section of two eccentrically placed cylinders with fluid between them to lubricate and prevent solid-to-solid contact. There are two important aspects to this flow which differ from the previous two flows: First, there are no inflows and outflows, and second, there are no domain corners to generate singularities in the stress-field.

The author's interest in this problem is, first, just that it is quite different from the other two. It will be interesting to observe the behavior of the numerical method here because it omits two puzzling aspects of memory-fluid problems: history-dependence at inlets, and stress-singularities of unknown character. There is also physical interest in this problem because the effects of fluid elasticity on the load-bearing capacity of the bearing

may be beneficial. It would be able to predict load bearing-capacity from measurable material properties, and numerical modelling may help to do so.

III. COMPUTATIONAL METHOD. The strain measure in the integrand of eq. (1) is assumed to be determined by a deformation gradient, $\mathbf{E}_0(\tau)$, just as in nonlinear elasticity. Only in the present case, the deformation gradient is assumed to be computable from a system of linear, non-constant coefficient ordinary differential equations along the path followed by each particle at which the stress is to be evaluated. The usual deformation gradients of large-strain elasticity can also be obtained from special cases of the following evolution equations:

$$\begin{aligned}\dot{\mathbf{x}}(\tau) &= \mathbf{v} \cdot \mathbf{x}(\tau) \\ \mathbf{x}(0) &= \mathbf{x}_0 \\ \dot{\mathbf{E}}_0(\tau) &= \mathbf{F}(\mathbf{x}(\tau), \nabla \mathbf{v} \cdot \mathbf{x}(\tau)) \mathbf{E}_0(\tau) \\ \mathbf{E}_0(0) &= \mathbf{I}\end{aligned}\tag{5}$$

The first two sets of equations determine the pathline (streamline) followed by a particle to bring it from its position, \mathbf{x} , at time τ in the past to its present position at the stress evaluation point, \mathbf{x}_0 . To evaluate the integrand of eq. (4), these equations are solved as an initial value problem in reverse time. This determines the non-constant coefficient in the evolution equation for the gradient, which is assumed to be a traceless matrix, \mathbf{F} . The common deformation gradient, $\frac{\partial \mathbf{x}(\tau)}{\partial \mathbf{x}_0}$, is obtained when \mathbf{F} is $\nabla \mathbf{v}$ itself.

The fundamental strategy of the current numerical method is to choose constant strain-rate finite elements: then the evolution equation is a constant-coefficient equation on each element. This strategy is enabled by a basic property of linear ODEs: If we define a deformation gradient, \mathbf{E}_{τ_1} , relative to time τ_1 by

$$\begin{aligned}\dot{\mathbf{E}}_{\tau_1} &= \mathbf{F} \mathbf{E}_{\tau_1} \\ \mathbf{E}_{\tau_1}(\tau_1) &= \mathbf{I}\end{aligned}\tag{6}$$

then the strain relative to the present time, evaluated at any earlier time is give by matrix multiplication:

$$\mathbf{E}_0(\tau) = \mathbf{E}_0(\tau_1) \mathbf{E}_{\tau_1}(\tau)\tag{7}$$

This provides interface conditions between finite elements, so that only constant-coefficient equations need be solved on each element: it turns out that such solutions are known analytically, as is the pathline and transit time along it 1 — 3.

Thus, given an estimate of the solution to the problem in terms of a velocity field, the integrand of eq. (4) can straight-forwardly be computed at each historical time. In the current method, this is used in conjunction with a specially devised Gaussian quadrature formulas to approximate σ' :

$$\int_{-\infty}^{\infty} \mathbf{S}_0(\tau) m(\tau) d\tau \approx \sum_{k=1}^{N_f} \omega_k \mathbf{S}_0(\tau_k)\tag{8}$$

With what we have thus far, the stress can be computed in any trial velocity field; to approximate the solution to eqs. (1) and (2), the usual Galerkin procedure can be followed in which the residual of eq. (1) is dotted into a test function, \mathbf{v}^h , drawn from the same space as the trial solutions, and the result is integrated over the problem domain. After integration by parts and replacement of the spatial integral by a numerical integral with points ξ_e and weights θ_e , we get something which looks like

$$\sum \theta_e \boldsymbol{\sigma}' \cdot \nabla \mathbf{v}^h - 2z(\nabla \cdot \mathbf{u}^h)(\nabla \cdot \mathbf{v}^h) - \rho(\mathbf{u} \cdot \nabla) \mathbf{u}^h \cdot \mathbf{v}^h - \mathbf{v}^h \cdot \mathbf{f}(\xi_e) = 0 \quad (9)$$

The pressure term of eq. (3) has been replaced by a penalty term [3] with penalty parameter z ; thus there are no explicit pressure unknowns, and the continuity equation (2) is satisfied to $O(z^{-1})$. Eq. (9) illustrates the $R = 0$ case; for nonzero R , the obvious modification of adding a Newtonian viscous term is made.

The important point to observe about eq. (9) is that to evaluate its residual, it is required to evaluate the stress at the points ξ_e by means already discussed. To complete the method, what is needed is a means of correcting estimates of the discrete solution, based on evaluation of the residual; Newton's method might be an example of such a procedure, but, as we shall see, this is not entirely straight-forward. The current algorithm employs the inverse Broyden method [1,2] to solve the discrete nonlinear equations. An important point to be made here is that, regardless of the choice of iterative scheme, the method outlined here is enormously costly in practice, because for a reasonably fine mesh, each evaluation of the stress-field values at the spatial integration points is a potentially formidable computation.

IV. FAST ALGORITHMS. The method outlined in the previous section applies to isothermal, incompressible flows in a fixed spatial domain. These restrictions are not essential; material compressibility and temperature dependence can be handled in very similar fashion if "artificial (historical) time" is introduced, in which either density or temperature are used to change the time variable along the pathlines in such a way that a traceless matrix in the evolution equation is obtained [5]. The transformation to artificial time does not in itself seem to be computationally costly, but these problems involve added levels of complexity to an already complicated solution procedure with additional fields and corresponding equations. The resulting phenomena are likely to be more intricate in detail and more nonlinear in character. A similar observation can be made about free-surface flows: a well-developed methodology exists [6] to solve such problems, which can be directly interfaced with the method outlined here, but this also certain to render the computations more formidable than they now are. With the current algorithm, computations on a mesh which is refined only to the extent which seems to be required to obtain acceptable accuracy, at a shear rate normally occurring in polymer processing, the computation of a steady solution can take as long as 40 minutes on a Cray 1-A. This must be reduced drastically if the algorithm is to be used routinely in scientific and engineering research, particularly if the more physically realistic enhancements mentioned above are to be added. The remainder of this paper will discuss several approaches to the reduction in computational cost which are currently being implemented or investigated by the author.

Vectorization of Linear Equation Solving. A variety of new computers have the capability of carrying out hardware vector operations: rearranging the computer code in such a way that the compiler can take advantage of this capability can result in substantial savings in computational cost. One part of a typical code where such savings have a good chance of being realized is in the solution of linear equations. Unfortunately, in the current algorithm, it is not expected that this can dramatically reduce the run time. Linear equations are solved in the nonlinear iteration scheme, but this appears to account for a small portion of the computational cost. The major portion of the calculation is carried out at the element level with small arrays or scalar quantities involved in resolving element boundary crossings and accumulating the deformation gradient by small matrix multiplication. Vectorization offers little hope of speeding up these calculations. On the other hand, it is expected that linear equation solving will begin to play a more and more important role with the planned enhancements to the code discussed earlier. The Jacobian terms corresponding to the thermal energy equation are easy to form; likewise the part of the Jacobian associated with inertial terms and the unknown free-surface transformation are easy to deduce. Also, active research is under way aimed at producing the Jacobian terms associated with the non-Newtonian viscous terms (see below).

In short, the future development of the code seems to point in the direction of an algorithm which has a large, unsymmetric, and possibly not banded matrix to factor at each one of dozens of possible iterates. The current iteration method has only one, banded, symmetric, positive-definite matrix to factor at the outset, and a back-substitution at each iteration (the unsymmetric Jacobian contribution of the inertial terms is left to the inverse updating scheme). It therefore seems appropriate to modify the code at the present time to take full advantage of vectorization, in order to make sure the linear equation solving phase remains in the background, as it should.

Adaptive Memory Quadrature. The area which seems to show most promise in reduction of the computational cost is that of the stress calculation at an individual stress evaluation point. There seem to be several possible approaches, the underlying strategy of all of them is to take advantage of the fact that the stresses are being evaluated in what is hoped will be a convergent sequence of velocity iterates. Particularly further along in the sequence, the previous iterate should be able to provide a guide to estimate how much computation is absolutely necessary at the next iteration.

Perhaps the most obvious way to do this is to use the previous iterate to determine what N_j of eq. (8) should be in the next iterate. It is observed that in some flows, very many fewer quadrature points are needed to accurately compute σ' than in other flows. The strategy will be to begin the iterations with a nominal number of points for each stress evaluation point and increase or decrease that number in succeeding iterations, based on adaptive criteria determined from previous iterations.

Jacobian Approximation. The reason that the present algorithm employs an updating scheme rather than a direct calculation of the Jacobian is that it is not a trivial matter to construct the Jacobian, or even write down a closed form expression for it. It is clear that the stress at a point can depend on velocities far from that point, and therefore

the Jacobian of the residual of eq. (9) cannot have the usual finite element band and/or sparsity structure. In ref. 7, an approximation scheme for the Jacobian is proposed. It is not clear at present whether this approximation, some other, or even an exact computation of the Jacobian is best (the latter may be possible to undertake — it is not clear at this time). But the work of ref. 7 shows clearly the complexity involved. The terms of the Jacobian contribution from the extra stress are computed by tracking along streamlines. The resulting Jacobian element matrices are not square: Their column dimension depends on the number of different elements the particle path passes through before the final integration point of eq. (8) is located. It appears that a frontal solution technique is called for in order to handle the resulting global matrix 7.

Pseudo-Dynamic Relaxation. The hope in computing the Jacobian is that the computational cost will be more than returned in improved convergence rate over the inverse Broyden algorithm afforded by Newton's method or modified Newton's method with Broyden updates. But the complexity of the Jacobian calculation is such that this may never be realized, and it is well worth the investigation of other improvements to the iterative solution of the nonlinear equations. One avenue currently being explored is that of "pseudo-dynamic relaxation." The problem is cast as a time-dependent problem and steady solutions are obtained by letting the transient phenomena die out. In the algorithm presented here, the transient behavior does not represent the true dynamics of the non-Newtonian fluid; the stress is computed in the current velocity field as if it had been a steady field for all time, hence the name "pseudo-dynamic." To do otherwise would involve complexities beyond what seems manageable at present, though implementation of the pseudo-dynamic algorithm does open the door for future exploration of true dynamic behavior.

One may easily verify that the steady-states of the pseudo-dynamic algorithm are the same as the steady states of the true dynamic algorithm. The reason for taking the pseudo-dynamic approach is to produce a different kind of steady-state iteration scheme, in which the damping of the high frequency modes in the pseudo-dynamic response can be controlled by choice of time-stepping method and pseudo-time step. The reason that this seems to be a worthwhile avenue to explore is suggested by recent work of Y. Renardy and M. Renardy 8. They found that with a certain spatial discretization of the linearized operator associated with the equations of motion of a Maxwell fluid in a shearing flow, there were apparently spurious eigenvalues extremely close to the right half-plane, evidently introduced by the discretization. If this were also a consequence of finite element discretization, there could be severe consequences for iterative methods which behave like temporal iteration schemes. It is hoped that by controlling the time step and parameters of the pseudo-dynamic time-stepping method, the damping of the high frequency modes associated with any spurious eigenvalues can be damped to produce nearer monotonic, more rapid convergence of the resulting iterative method. There is no worry here of damping out interesting transient behavior — the transient behavior is not correct, and all that it is required is that it be damped out as rapidly as possible.

The following algorithm is based on Hughes, Liu, and Brooks predictor-corrector algorithm for the Navier-Stokes equations 9, but with the possibility of more fully implicit

inner iterations at each time step:

$$\begin{aligned}
 \mathbf{M}\mathbf{a}_{n+1} + \mathbf{C}\mathbf{v}_{n+1} + \mathbf{N}(\mathbf{v}_{n+1}) + \mathbf{Q}(\mathbf{v}_{n+1}) &= \mathbf{F}_{n+1} \\
 \mathbf{Q} &= \int_{\Omega} \mathbf{B}^T \boldsymbol{\sigma} dV - \mathbf{C}\mathbf{v} \\
 \mathbf{N} &= \text{nonlinear inertial term, excl. time deriv.} \\
 \mathbf{v}_{n+1}^{(0)} &= \mathbf{v}_n + (1 - \gamma)\Delta t \mathbf{a}_n \\
 \mathbf{v}_{n+1}^{(i+1)} &= \mathbf{v}_{n+1}^{(i)} - \mathbf{J}^{-1} \{ [\mathbf{M} + \gamma\Delta t \mathbf{C}] \mathbf{v}_{n+1}^{(i)} + \gamma\Delta t \mathbf{Q}(\mathbf{v}_{n+1}^{(i)}) \\
 &\quad + \gamma\Delta t \mathbf{N}(\mathbf{v}_{n+1}^{(i)}) - \mathbf{M}\mathbf{v}_{n+1}^{(0)} + \gamma\Delta t \mathbf{F}_{n+1} \} \\
 \mathbf{a}_{n+1} &= (\mathbf{v}_{n+1} - \mathbf{v}_{n+1}^{(0)}) / \gamma\Delta t \\
 \mathbf{J} &= \mathbf{M} + (\gamma\Delta t \frac{\partial \mathbf{Q}}{\partial \mathbf{v}})_{opt} + (\gamma\Delta t \frac{\partial \mathbf{N}}{\partial \mathbf{v}})_{opt}
 \end{aligned} \tag{10}$$

\mathbf{M} is the finite element "mass matrix," \mathbf{C} the Newtonian viscous and penalty-pressure matrix, and \mathbf{F}_{n+1} the applied force vector at time step $n + 1$. \mathbf{B} is the usual finite element matrix of shape function derivatives and $\boldsymbol{\sigma}$ is the stress, computed in $\mathbf{v}_{n+1}^{(i+1)}$ as described in previous sections; \mathbf{v}_{n+1} without the superscript of inner iteration is the "fully converged" result of inner iteration at time-level $n + 1$. Choice of the number of inner iterations is open, so that \mathbf{v}_{n+1} could result from just one correction cycle, or many. An important aspect of eq. (10) is found in those terms labelled by $(\cdot)_{opt}$; a fully implicit treatment would employ exact Jacobian terms here. At the other extreme is Hughes, Liu, and Brooks method; they use \mathbf{C} to approximate both of these terms and do only one inner iteration. The present non-Newtonian implementation uses \mathbf{C} initially, updated by the inverse Broyden method during a number of inner iterations. If it proves to be effective, Newton or modified Newton/Broyden iterations could be employed in the inner iterations. It is instructive to note that the direct steady-state Broyden algorithm mentioned earlier is obtained as a special case of eq. (10) with $\gamma = 1$ and an infinite time step.

V. SUMMARY. A pilot numerical method for the computation of solution to memory fluid flow problems has been described. This method has shown that such computations are feasible but extremely costly. More reasonable physical assumptions than those of isothermal, incompressible flow in a fixed domain are on the near horizon but are bound to increase the computational cost. A number of ways of improving the computational performance of the algorithm have been proposed here and are in the implementation stage. These improvements will go together to make what the author refers to as "a fast algorithm for non-Newtonian flow." It is hoped that this fast algorithm can transform the method described here from pilot code to useful computational tool for the investigation of problems in viscoelasticity and rheology.

Acknowledgements: The numerical techniques for integral constitutive equations described here were developed jointly by the author and B. Bernstein (Dept. Mathematics, I. I. T.).

REFERENCES.

1. D. S. Malkus, finite element methods for viscoelastic flow, MRC Technical Summary Report No. 2812, Mathematics Research Center, University of Wisconsin, Madison, WI (1985).
2. D. S. Malkus and B. Bernstein, Flow of a Curtiss-Bird fluid over a transverse slot using the finite element drift-function method, *J. Non-Newtonian Fluid Mechs.* 16, 77(1984).
3. B. Bernstein, D. S. Malkus, and E. T. Olsen, A finite element for incompressible plane flows of fluids with memory, *Int. J. Numer. Meths. Fluids* 5, 43(1985).
4. B. Bernstein, M. K. Kadivar, and D. S. Malkus, Steady flows of memory fluids with finite elements: Two test problems, *Comp. Meths. Appl. Mechs. Eng.* 27, 279 (1981).
5. B. Bernstein, private communication.
6. H. S. Kheshgi and L. E. Scriven, Penalty finite element analysis of time-dependent two-dimensional free surface film flows, in *Finite Element Flow Analysis*, Ed: T. Kawai, University of Tokyo Press, Tokyo (1982).
7. G. Zazi, Ph. D. Thesis, Illinois Institute of Technology, Chicago, IL (1985).
8. Y. Renardy and M. Renardy, Trans. 3rd Army Conf. on Appl. Math. and Computing, this volume.
9. T. J. R. Hughes, W. K. Liu, and A. Brooks, Finite element analysis of incompressible viscous flows by the penalty function formulation, *J. Comp. Phys.* 30, 1 (1979).

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER #2860	2. GOVT ACCESSION NO. AD-A160 877	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A FAST ALGORITHM FOR NON-NEWTONIAN FLOW		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David S. Malkus		8. CONTRACT OR GRANT NUMBER(s) AFOSR-85-0141, MCS83-01433 DAAG29-80-C-0041 MCS 79-03542, 81-02089
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of 610 Walnut Street Wisconsin Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Numbers 2 & 3 - Physical Math. & Numerical Anal. & Scientific Computing
11. CONTROLLING OFFICE NAME AND ADDRESS (See Item 18 below)		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 10
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES U. S. Army Research Office Air Force Office of National Science Foundation P. O. Box 12211 Scientific Research Washington, DC 20550 Research Triangle Park Washington, DC 20332 North Carolina 27709		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) memory fluid, non-Newtonian fluid, finite element, vectorization, adaptive memory quadrature, Jacobian, pseudo-dynamic relaxation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The goals of the project described here are twofold: First, to turn an existing pilot algorithm for the steady flow of non-Newtonian memory fluids into a robust and efficient algorithm. Second, render enhancements of the method's current capabilities computationally feasible. Such enhancements include fully coupled thermal dependence, material compressibility, and free surface flows. The pilot algorithm is a finite element method whose novelty lies in its computation of the stress field in a nonlinear iteration scheme. The stress at a point is a non-local functional of the current velocity iterate,		

20. Abstract

and the pilot method has demonstrated the feasibility of reliable computation with such constitutive equations. Before the method can take its place as a reliable scientific and engineering tool, intensive effort must be made to reduce the computational cost in the manner described here.

END

FILMED

12-85

DTIC